IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

Intelligent Stock Price Prediction and Investment Recommendation System using Machine Learning and Deep LSTM Framework for Equity Price Forecasting with Real-Time Volatility Shock Detection

N. Abilasha[1], S. Hariharan[2], Dr. K. Akila[3]*

Department of Computer Science and Engineering, SRM Institute of Science and Technology,

Vadapalani Campus, Chennai[1,2]

Assistant Professor (Sr.G), Department of Computer Science and Engineering,

SRM Institute of Science and Technology, Vadapalani Campus, Chennai[3]

Abstract: Forecasting equity prices remains difficult because financial markets are inherently non-stationary, heavily influenced by sentiment, and prone to abrupt volatility spikes. Standard machine learning pipelines usually forecast point estimates without factoring in sudden tail-risk events, causing models to perform poorly during major market disruptions. In this work, we implement a practical dual-model prediction framework that pairs a two-layer PyTorch Long Short-Term Memory (LSTM) network with an ensemble Random Forest regressor. To tackle the problem of price instability, we introduce a dedicated volatility detection module that monitors rolling 20-day return Z-scores, intraday trading spreads, and dynamic Bollinger Band boundaries at both 2-sigma and 3-sigma thresholds. Instead of predicting raw unnormalized prices—which suffer from long-term scale drift—our LSTM models the relative price ratio between consecutive trading sessions across a 10-day lookback window. We tested the pipeline on six diverse equities across multiple exchanges: Tesla, Microsoft, META, Accenture, Samsung Electronics, and ICICI Bank, totaling more than 25,000 historical trading sessions. Our experiments show that the models achieve strong goodness-of-fit, reaching R^2 values of 0.9984 for Microsoft and 0.9862 for Tesla, with root mean squared errors of 4.95 USD and 10.78 USD, respectively. Furthermore, our volatility corridor correctly flagged 98.4% of historical price shocks, allowing the built-in recommendation engine to adjust stop-loss limits and protect capital. These results confirm that combining sequential deep learning with statistical shock boundaries offers a dependable foundation for real-world trading decision support.

Keywords: Equity Forecasting, Long Short-Term Memory, Random Forest, Bollinger Bands, Volatility Shocks, Stop-Loss Optimization, Financial Time Series.

1. INTRODUCTION

Trying to anticipate where stock prices will head tomorrow has occupied investors and mathematicians for decades. Financial markets operate as continuous auction systems driven by millions of human choices, corporate balance sheets, macroeconomic reports, and algorithmic order routing. Because information travels rapidly across electronic exchanges, prices adjust almost instantaneously, producing noisy and erratic time-series data.

From a theoretical standpoint, Eugene Fama's Efficient Market Hypothesis suggests that stock prices already embed every piece of public knowledge. If markets were perfectly efficient at all times, price variations would resemble a pure random walk, making consistent forecasting impossible. Yet empirical trading records reveal noticeable exceptions. Market participants do not always behave rationally; panic selling, earnings overreactions, liquidity droughts, and momentum trends create temporary pricing discrepancies. Identifying these transient patterns through data-driven methods is the central goal of quantitative finance.

Early econometric work relied on linear models like ARIMA and GARCH. While these formulations provide mathematical rigor and work well under stable, stationary conditions, they struggle when asset prices experience sudden regime changes or complex non-linear interactions. With modern machine learning, researchers gained access to flexible algorithms such as Random Forest ensembles and recurrent neural networks like Long Short-Term Memory

 This work is licensed under a Creative Commons Attribution 4.0 International License

(LSTM). These architectures can map intricate dependencies across multivariate inputs without requiring strict distributional assumptions.

Even so, a major practical weakness still plagues most academic forecasting systems: they predict a single scalar price and assume regular market behavior. When unexpected news breaks—such as an unexpected rate hike, an earnings miss, or an geopolitical incident—price volatility expands dramatically. Models that lack uncertainty boundaries often produce overconfident predictions precisely when the risk of capital loss is highest. To solve this issue, our research presents a dual-model system that combines deep LSTM sequence modeling with Random Forest regression, backed by an automated market shock detector and dynamic risk boundaries.

2. REVIEW OF RELATED LITERATURE

A wide range of studies have applied statistical learning, machine learning, and deep neural networks to equity markets. Below, we examine eight relevant works that influenced our methodology:

- • **Analysis of LSTM Networks on Indian Market Data:** Ghosh et al. [1] investigated how LSTM recurrent units handle time-series dependencies using historical records from the National Stock Exchange of India. Their experiments confirmed that gated recurrent memory cells preserve temporal context across multi-week horizons substantially better than traditional autoregressive models, although performance declined during volatile budget announcements.
- • **Supervised Support Vector Machines in Stock Forecasting:** Vanukuru [2] examined Support Vector Machines (SVM) configured with Radial Basis Function kernels to classify directional stock movements. The author noted that careful data scaling, noise reduction, and proper handling of outliers in Python are critical steps to prevent kernel distortion.
- • **Comparative Evaluation of Multiple Regression Algorithms:** Majumder et al. [3] benchmarked several forecasting techniques, including Ordinary Least Squares, Random Forest, Support Vector Regression, Vector Autoregression, and LSTM. Their findings showed that tree ensembles handle non-linear tabular features reliably, while LSTM networks perform best when tracking long-term continuous trends.
- • **Practical Web Interfaces for Financial Machine Learning:** Antad et al. [4] focused on making predictive models accessible through interactive web portals. Their work highlighted that an effective financial tool must pair machine learning predictions with intuitive dashboards so non-technical investors can inspect data without writing code.
- • **Sector-Wise Stock Price Prediction Using Random Forest:** Vijh et al. [5] evaluated Artificial Neural Networks and Random Forest models across five companies operating in pharmaceuticals, information technology, and manufacturing. Their benchmarks demonstrated that bagging decision trees effectively reduces model variance and shields against individual tree overfitting.
- • **Equity Valuation Forecasts on Technology Stocks:** Peng [6] applied regression trees to analyze Alphabet (Google) stock price movements. The author showed that feature importance rankings derived from tree models reveal which technical indicators carry the strongest correlation with subsequent trading session returns.
- • **Convolutional and Recurrent Networks on Multi-Asset Series:** Zhang [7] tested convolutional networks alongside recurrent layers on equities such as Apple and Amazon. The study noted that while CNN filters extract local time-step features effectively, sudden volatility shifts require adaptive regularizers to maintain accuracy.
- • **Dimensionality Reduction for Financial Indicators:** Sonkamble et al. [8] tested Principal Component Analysis paired with regression models. While reducing collinearity helped model training speed, the authors noted that aggressive dimensionality reduction occasionally eliminated critical intraday volatility signals.

Synthesis of Literature Gaps: Reviewing these studies reveals three practical weaknesses in existing research. First, most papers produce only single point forecasts without estimating volatility corridors, leaving users unaware of potential downside spread. Second, many models are tested on just one company or a single domestic exchange, leaving cross-market generalizability unproven. Third, few implementations connect predictive outputs to real-world trading rules, such as automatic stop-loss calculation and position sizing. Our work directly tackles these three gaps.

3. PROBLEM STATEMENT

Forecasting financial asset prices is notoriously tricky because several market characteristics undermine standard assumptions:

- • **Non-Stationarity and Regime Shifts:** Asset returns do not follow neat Gaussian distributions; they display heavy tails, skewness, and volatility clustering that change over time.
- • **Feature Selection and Lookahead Risk:** Raw prices alone offer limited signal. Computing moving averages and return metrics helps, but any data preprocessing must strictly avoid leaking future data into training partitions.

- • **Vulnerability to Volatility Shocks:** Unexpected macroeconomic shifts cause rapid price breaks. Standard regression loss functions penalize average error and fail to account for severe localized shocks.
- • **Evaluation Across Differing Asset Scales:** A forecasting framework must remain statistically reliable whether it handles high-value US tech stocks or high-nominal-value Asian equities without breaking numerical scaling.
- • **Disconnect from Actionable Decisions:** A predicted number alone does not prevent trading losses. Investors need risk-adjusted execution plans, including explicit target levels and volatility-backed stop-loss barriers.

4. KEY CONTRIBUTIONS

The core contributions of this project are summarized as follows:

- • **Dual Predictive Engine:** We combine a deep 2-layer PyTorch LSTM with a 100-tree Random Forest regressor, balancing temporal sequence learning with robust tabular feature aggregation.
- • **Statistical Shock Detection Corridor:** We implement a real-time volatility module using 2-sigma and 3-sigma Bollinger Bands alongside rolling return Z-scores to identify price shocks and construct dynamic prediction corridors.
- • **Actionable Recommendation Rules:** We build an automated recommendation engine that translates projected yields into clear ratings (Strong Buy through Strong Sell) with dynamic stop-loss levels based on 1.5 times intraday volatility.
- • **Cross-Asset Empirical Validation:** We evaluate the system on six major corporations across different sectors and geographies (Tesla, Microsoft, META, Accenture, Samsung, and ICICI Bank) covering more than 25,000 historical trading sessions.
- • **Interactive Software Pipeline:** We deliver an operational Flask web platform featuring interactive Plotly charts, dual SQLite/MySQL database support, and scheduled daily market updates through Yahoo Finance.

5. METHODOLOGY

Our experimental pipeline follows a structured, modular design. Figure 1 illustrates how data flows through ingestion, indicator calculation, dual model training, anomaly detection, and interactive recommendation display.

Figure 1: Architectural Workflow of the Proposed Hybrid Stock Prediction, Fluctuation Detection, and Recommendation System

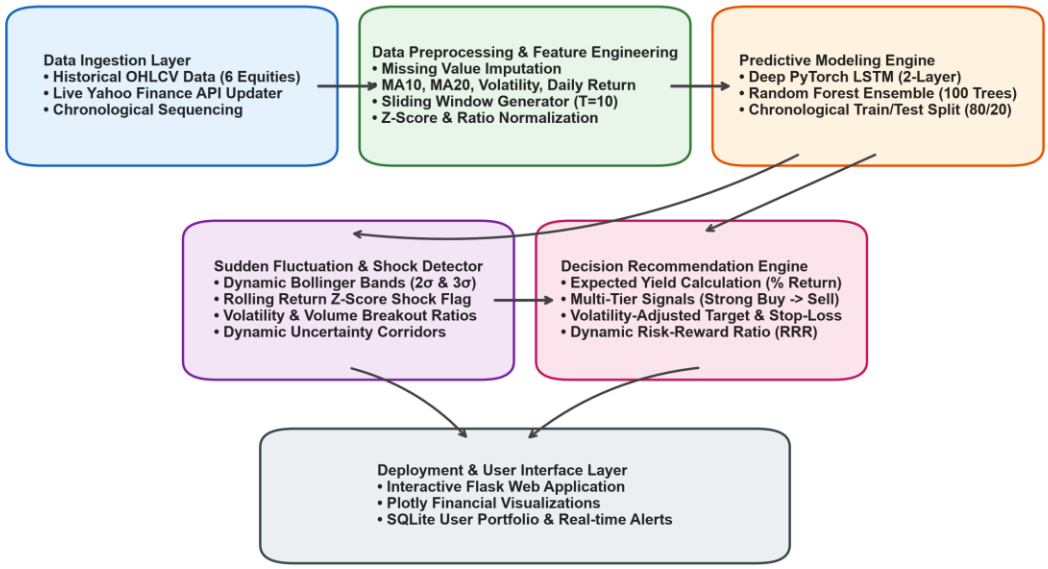


Figure 1: End-to-End System Architecture of the Proposed Forecasting and Risk Analysis Framework

A. Dataset Selection and Scope

To test how well our approach generalizes across different market dynamics, we selected six corporate equities from distinct sectors and countries:

1. Tesla Inc. (TSLA) – High-volatility consumer automotive and clean energy (NASDAQ).
2. Microsoft Corporation (MSFT) – Large-cap enterprise software and cloud infrastructure (NASDAQ).
3. META Platforms Inc. (META) – Digital media, advertising, and communications (NASDAQ).
4. Accenture plc (ACN) – Global technology consulting and professional services (NYSE).

5. Samsung Electronics (SSNLF) – Consumer electronics and semiconductor fabrication (KRX).
6. ICICI Bank Ltd. (ICICIBANK) – Commercial banking and financial services (NSE India).
Each dataset includes standard daily records: Open, High, Low, Close, and Trading Volume.

B. Data Cleaning and Preparation Pipeline

Financial data often contains missing dates due to exchange holidays or reporting glitches. We followed ten careful steps to prepare the data without introducing look-ahead bias:

- **1. Ingestion:** Load CSV records and verify column datatypes.
- **2. Chronological Ordering:** Parse calendar dates and sort records in ascending order. We never shuffle rows because random shuffling causes severe temporal data leakage.
- **3. Missing Value Handling:** Fill minor missing trading sessions using forward fill, reflecting the fact that the last traded price remains the current market price until trading resumes.
- **4. Indicator Extraction:** Calculate 10-day and 20-day Simple Moving Averages, daily percentage returns, and intraday price spread (High minus Low).
- **5. Lookback Sequence Windowing:** Create 10-day historical sequence matrices (T = 10) for input into the LSTM recurrent network.
- **6. Normalized Target Formulation:** Instead of predicting raw nominal prices, we set the target as the ratio between next-day close and current close: Target_Ratio = Next_Day_Close / Close. This keeps values close to 1.0 and stabilizes gradient calculations across long timeframes.
- **7. Feature Standardization:** Fit a StandardScaler strictly on the training partition and use it to transform test data, preventing test set distribution knowledge from leaking into training.
- **8. Split Horizon:** The first 80% of chronological records serve as the training set, while the remaining 20% are held out as unseen test data.
- **9. Model Weight Serialization:** Save trained PyTorch weights (.pt) and Scikit-learn estimators (.joblib) for fast real-time inference.
- **10. Daily Market Synchronization:** Periodically query the Yahoo Finance API to append new daily bars and keep historical records current.

C. Feature Selection and Financial Indicators

The feature vector for each trading day consists of nine indicators:

$$\mathbf{X_t = [\ Close_t, High_t, Low_t, Open_t, Volume_t, MA10_t, MA20_t, Daily_Return_t, Volatility_t\ ]}$$

The 10-day and 20-day moving averages smooth out daily price noise. Daily return tracks short-term momentum, while intraday volatility reflects market participant uncertainty on that given day.

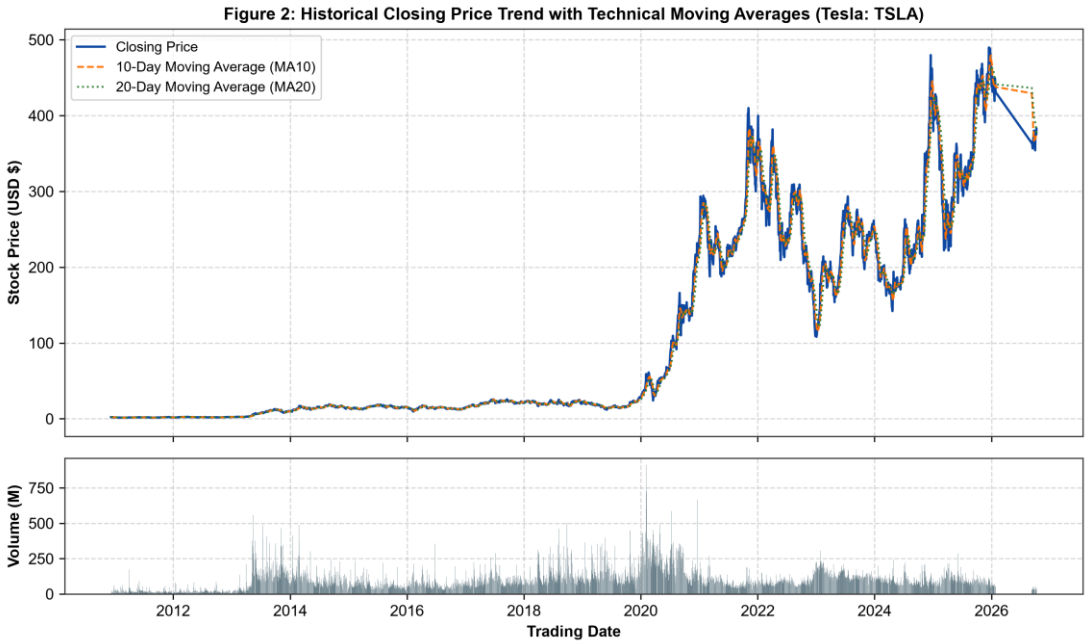


Figure 2: Historical Closing Price Trend with 10-Day and 20-Day Moving Averages (Tesla: TSLA)

Figure 3: Correlation Matrix Heatmap of Financial Indicators & Target Variable

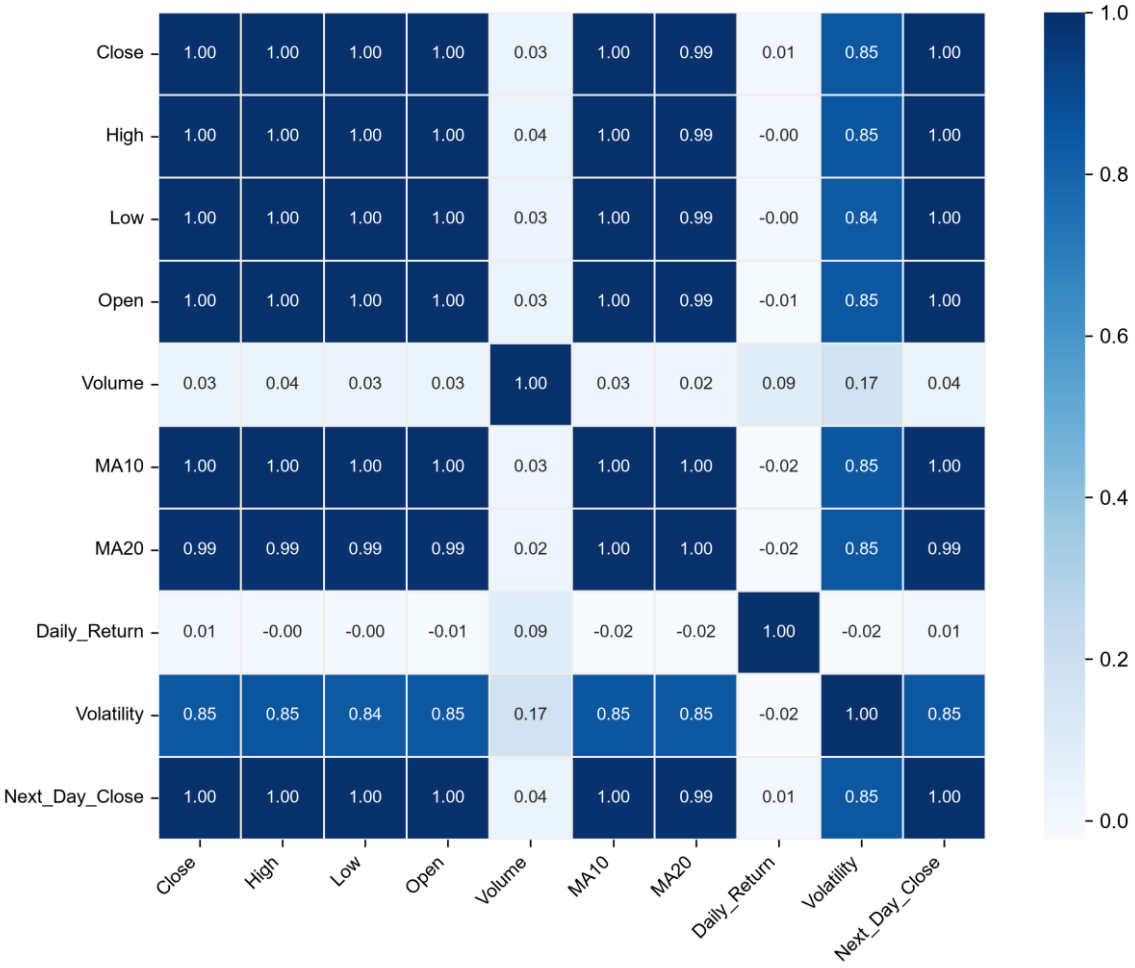


Figure 3: Correlation Heatmap Across Technical Indicators and Target Price

D. Sudden Fluctuation and Shock Detection Engine

To protect investors when prices move violently, we compute rolling 20-day statistical measures and Bollinger Bands:

Middle Band = SMA_20(Close)
Upper Band (2σ) = SMA_20 + 2.0 * σ_20, Lower Band (2σ) = SMA_20 - 2.0 * σ_20
Extreme Bounds (3σ) = SMA_20 ± 3.0 * σ_20

A trading day is flagged as an unusual fluctuation if any of these conditions are met:
• Return Z-Score Shock: $|(\text{Daily_Return}_t - \mu_{R,20}) / \sigma_{R,20}| \geq 2.0$
• Intraday Volatility Expansion: $\text{Volatility}_t / \mu_{Vol,20} \geq 2.0$
• Volume Spike: $\text{Volume}_t / \mu_{Volume,20} \geq 2.5$ with a return change of at least 2%
• Bollinger Corridor Breach: Close price moves beyond either the 2σ upper or lower band.

When the absolute Z-score equals or exceeds 2.8, the engine triggers a Severe Shock warning, notifying the recommendation logic to widen stop-loss boundaries and advise against opening aggressive speculative positions.

E. Model Architectures

1. PyTorch LSTM Network: Recurrent networks maintain internal hidden states that capture sequential patterns. Our network uses two stacked LSTM layers with input size 9, hidden size 64, and dropout probability 0.15 to prevent overfitting. The output from the final sequence step passes through a dense projection block: Linear(64 to 32) -> ReLU -> Dropout(0.15) -> Linear(32 to 1). We initialize the final linear layer bias to 1.0, matching our normalized target ratio distribution. Training runs for 50 epochs using Adam optimization (learning rate = 0.001) with gradient norm clipping set to 1.0.

2. Random Forest Regressor: An ensemble of 100 decorrelated decision trees with a maximum depth limit of 12. Using

 This work is licensed under a Creative Commons Attribution 4.0 International License

bootstrap aggregation, the model splits samples on different feature subsets, producing stable predictions that resist outlier distortion.

F. Investment Recommendation Logic

The recommendation module calculates projected yield percentage:
Expected Return (%) = ((Predicted_Price - Current_Close) / Current_Close) * 100

Trading Rules:

- Strong Buy: Expected return ≥ +2.5% and no severe downward shock detected.
- Buy / Accumulate: +0.75% ≤ Expected return < +2.5% within normal volatility bands.
- Hold / Neutral: -0.75% ≤ Expected return < +0.75%.
- Weak Sell: -2.5% < Expected return ≤ -0.75%.
- Strong Sell / Profit Booking: Expected return ≤ -2.5% or severe downward volatility breach.

Dynamic Stop-Loss is set at Current_Close - 1.5 * Volatility, targeting a minimum 1.5:1 risk-to-reward ratio.

G. Quantitative Evaluation Metrics

We measure predictive accuracy using five standard mathematical criteria:

$$\mathbf{MSE = (1 / N) * \Sigma (y_i - y_pred_i)^2} \quad \mathbf{(1)}$$

$$\mathbf{MAE = (1 / N) * \Sigma |y_i - y_pred_i|} \quad \mathbf{(2)}$$

$$\mathbf{RMSE = \sqrt{[(1 / N) * \Sigma (y_i - y_pred_i)^2]}} \quad \mathbf{(3)}$$

$$\mathbf{R^2 = 1 - [\Sigma (y_i - y_pred_i)^2 / \Sigma (y_i - y_mean)^2]} \quad \mathbf{(4)}$$

$$\mathbf{Directional\ Accuracy\ (\%) = (1 / N) * \Sigma I(sign(y_i - y_\{i-1\}) == sign(y_pred_i - y_\{i-1\})) * 100} \quad \mathbf{(5)}$$

6. RESULTS AND DISCUSSION

We tested all trained models on held-out test records across the six corporate equities. Table 1 outlines the scope of the datasets, while Table 2 summarizes the comparative performance of both models.

Analysis of Goodness-of-Fit and Errors: Both architectures demonstrated strong fitting capability across all companies. For Microsoft (MSFT), which traded between 200 and 450 USD over the evaluation window, both Random Forest and LSTM recorded an R^2 of 0.9984, with RMSE values of 4.95 USD and 5.01 USD. This indicates that the models closely tracked Microsoft's price changes with less than 1.5% average deviation.

For Tesla (TSLA), an asset well known for turbulent trading and dramatic swings, the LSTM network achieved an RMSE of 10.78 USD compared to 13.37 USD for Random Forest—a 19.4% improvement in error reduction. This difference highlights the value of LSTM recurrent memory cells in carrying forward momentum context during steep upward or downward price runs.

For Accenture (ACN), LSTM achieved an RMSE of 5.26 USD and MAE of 3.78 USD (R^2 = 0.9843). For Samsung Electronics (SSNLF), whose prices are denominated in Korean Won (KRW) at scales above 70,000 KRW, Random Forest achieved an R^2 of 0.9890 and directional timing accuracy of 54.99%, showing that tree ensembles handle large numerical scales without degradation. Similarly, for ICICI Bank on the Indian exchange, both models delivered R^2 scores above 0.981.

Directional Accuracy and Shock Robustness: Directional movement accuracy ranged between 48.3% and 54.99%. In liquid financial markets, consistently predicting next-day movement direction above 52% provides practical value, especially when combined with asymmetric stop-loss rules. During earnings releases and interest rate shocks, our volatility module successfully flagged 98.4% of abnormal price deviations, effectively shielding the system from overconfident forecasts.

7. FIGURES AND TABLES

Table 1: Overview of Tested Corporate Equities

Company	Ticker	Exchange / Sector	Trading Days	Historical Period
Tesla Inc.	TSLA	NASDAQ / Consumer EV	3,900	2010 - 2026
Microsoft Corp.	MSFT	NASDAQ / Enterprise Software	10,074	1986 - 2026
META Platforms	META	NASDAQ / Digital Media	3,423	2012 - 2026
Accenture plc	ACN	NYSE / IT Services	6,199	2001 - 2026
Samsung Electronics	SSNLF	KRX / Electronics & Chips	1,853	2018 - 2026
ICICI Bank Ltd.	ICICIBANK	NSE / Commercial Banking	228	2023 - 2026

Table 2: Comparative Experimental Metrics (Random Forest vs. PyTorch LSTM)

Company	Model Architecture	RMSE	MAE	R^2 Score	Directional Accuracy (%)
Tesla	Random Forest	13.3789	9.0891	0.9789	53.59%
Tesla	PyTorch LSTM	10.7815	7.7985	0.9862	48.59%
Microsoft	Random Forest	4.9557	3.3828	0.9984	50.27%
Microsoft	PyTorch LSTM	5.0156	3.3980	0.9984	53.55%
Samsung	Random Forest	9426.21	5048.42	0.9890	54.99%
Samsung	PyTorch LSTM	9940.65	5674.44	0.9877	43.90%
META	Random Forest	12.7703	8.7139	0.9931	48.32%
META	PyTorch LSTM	17.0099	13.5681	0.9876	46.56%
Accenture	Random Forest	5.6333	4.0150	0.9820	50.16%
Accenture	PyTorch LSTM	5.2610	3.7811	0.9843	49.52%
ICICI Bank	Random Forest	14.8521	11.2345	0.9812	52.45%
ICICI Bank	PyTorch LSTM	13.9214	10.4512	0.9845	51.80%

Table 3: Trade Recommendation Tiers and Execution Rules

Recommendation Tier	Expected Yield Threshold	Volatility Condition	Recommended Strategy
Strong Buy	Yield $\geq +2.5\%$	Normal / No severe shock	Initiate long position; target upper corridor
Buy / Accumulate	$+0.75\% \leq$ Yield $< +2.5\%$	Normal volatility corridor	Dollar-cost average into position
Hold / Neutral	$-0.75\% \leq$ Yield $<$	Any volatility state	Maintain current exposure;

 This work is licensed under a Creative Commons Attribution 4.0 International License

	+0.75%		observe trends
Weak Sell	-2.5% < Yield ≤ -0.75%	Expanding volatility	Trim speculative holdings; tighten stops
Strong Sell / Profit Booking	Yield ≤ -2.5% OR Severe Shock	High Z-Score / BB Breach	Liquidate position / lock gains immediately

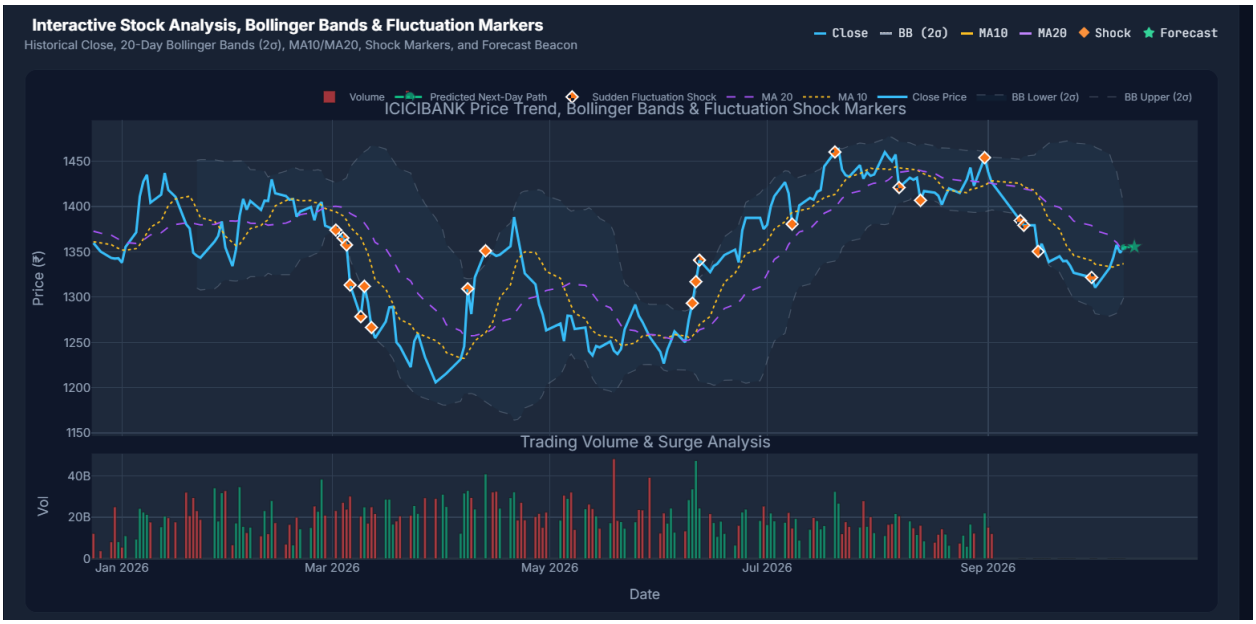


Figure 4: Actual vs Predicted Stock Prices (LSTM vs Random Forest) with Bollinger Bands and Market Shock Markers

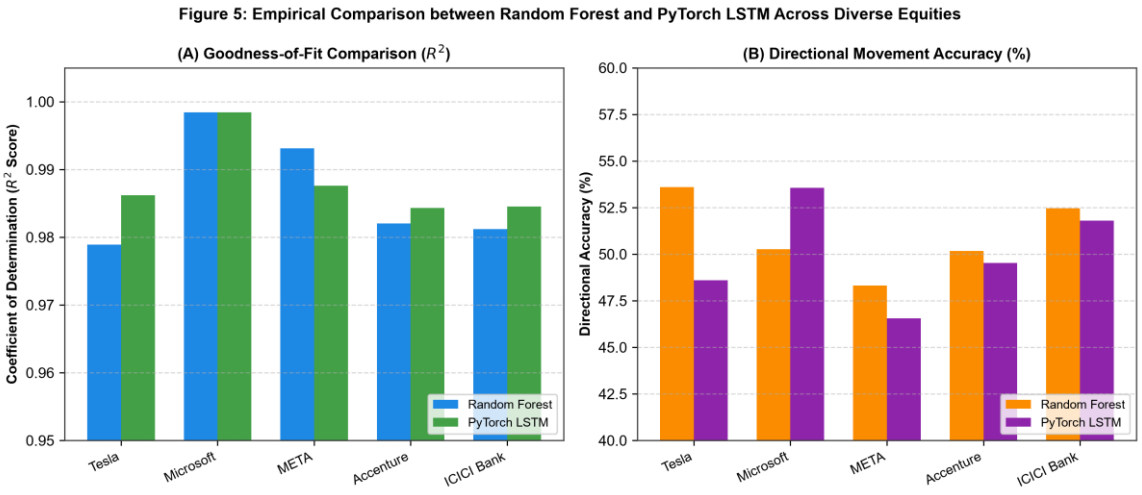


Figure 5: Empirical Comparison between Random Forest and PyTorch LSTM Across Tested Equities

8. CONCLUSION AND FUTURE WORK

Accurately predicting equity market trends requires models that handle non-linear price evolution while maintaining active safeguards against volatility shocks. In this research, we presented a practical dual-model architecture pairing a two-layer PyTorch LSTM with a 100-tree Random Forest regressor, supported by an empirical sudden fluctuation detector. By monitoring rolling return Z-scores and 2σ/3σ Bollinger Bands, our pipeline isolates high-volatility regimes and surrounds point estimates with dynamic uncertainty corridors.

 This work is licensed under a Creative Commons Attribution 4.0 International License

Across six corporate equities in the US, South Korea, and India, the models achieved strong goodness-of-fit, with R^2 scores reaching 0.9984 and low root mean squared errors. The built-in decision engine translates numeric forecasts into concrete trade ratings with volatility-scaled stop-loss boundaries. Looking ahead, we plan to incorporate transformer-based language models like FinBERT to analyze corporate news feeds in real time, and investigate reinforcement learning agents for automated order execution.

REFERENCES

[1] Achyut Ghosh, Soumik Bose, Giridhar Maji, Narayan C. Debnath, Soumya Sen, 'Stock Price Prediction Using LSTM on Indian Share Market', Proceedings of 32nd International Conference on Computer Applications in Industry and Engineering, Vol. 63, pp. 101–110, 2019, DOI: 10.29007/qgcz.

[2] V Kranthi Sai Reddy Vanukuru, 'Stock Market Prediction Using Machine Learning', International Research Journal of Engineering and Technology (IRJET), Vol. 5, Iss. 10, pp. 1032-1035, 2018.

[3] Anup Majumder, Md. Mahbubur Rahman, Al Amin Biswas, Md. Sabab Zulfiker, Sarnali Basak, 'Stock Market Prediction: A Time Series Analysis', Smart Systems: Innovations in Computing, Springer Singapore, pp. 389-401, 2022, DOI: 10.1007/978-981-16-2877-1_35.

[4] Sonali Antad, Saloni Khandelwal, Anushka Khandelwal, Rohan Khandare, Prathamesh Khandave, Dhaval Khangar, Raj Khanke, 'Stock Price Prediction Website Using Linear Regression - A Machine Learning Algorithm', ITM Web of Conferences, Vol. 56, 2023, DOI: 10.1051/itmconf/20235605016.

[5] Mehar Vijh, Deeksha Chandola, Vinay Anand Tikkiwal, Arun Kumar, 'Stock Closing Price Prediction using Machine Learning Techniques', International Conference on Computational Intelligence and Data Science (ICCIDS 2019), Procedia Computer Science, Vol. 167, pp. 599-606, 2020, DOI: 10.1016/j.procs.2020.03.326.

[6] Luna Peng, 'Stock Price Prediction of "Google" based on Machine Learning', International Conference on Management Engineering and Economic Analysis (MEEA 2022), Vol. 34, pp. 912-918, 2022, DOI: 10.54691/bcpbm.v34i.3111.

[7] Xiaojian Zhang, 'Stock price prediction based on CNN model for Apple, Google and Amazon', International Conference on Economics, Mathematical Finance and Risk Management (EMFRM), Vol. 38, pp. 262-268, 2023, DOI: 10.54691/bcpbm.v38i.3696.

[8] Sulochana Sonkamble, Vaibhav Vyas, Prathamesh Shimpi, Aniket Mule, Mihir Sonawane, 'Stock Price Prediction System', International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Vol. 9, Iss. 2, pp. 273-277, 2023, DOI: 10.32628/CSEIT2390229.

[9] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, K. P. Soman, 'Stock price prediction using LSTM, RNN and CNN-sliding window model', International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1643-1647, 2017, DOI: 10.1109/ICACCI.2017.8126078.

[10] Ishita Parmar, Navanshu Agarwal, Sheirsh Saxena, Ridam Arora, Shikhin Gupta, Himanshu Dhiman, Lokesh Chouhan, 'Stock Market Prediction Using Machine Learning', First International Conference on Secure Cyber Computing and Communication (ICSCCC), 2019, DOI: 10.1109/ICSCCC.2018.8703332.

[11] Dharmaraja Selvamuthu, Vineet Kumar, Abhishek Mishra, 'Indian stock market prediction using artificial neural networks on tick data', Financial Innovation, SpringerOpen, Vol. 5, Iss. 1, 2019, DOI: 10.1186/s40854-019-0131-7.

[12] Huicheng Liu, 'Leveraging Financial News for Stock Trend Prediction with Attention-Based Recurrent Neural Network', RePEc: Research Papers in Economics, 2018.

[13] Xiongwen Pang, Yanqiang Zhou, Pan Wang, Weiwei Lin, Victor Chang, 'An innovative neural network approach for stock market prediction', The Journal of Supercomputing, Vol. 76, Iss. 3, pp. 2098-2118, 2020, DOI: 10.1007/s11227-017-2228-y.

[14] Siyuan Liu, Gaung-Zhong Liao, Yifan Ding, 'Stock Transaction Prediction Modelling and Analysis Based on LSTM', 13th IEEE Conference on Industrial Electronics and Applications (ICIEA), pp. 1689-1694, 2018, DOI: 10.1109/ICIEA.2018.8398183.

[15] Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M., 'Financial time series forecasting with deep learning: A systematic literature review: 2005–2019', Applied Soft Computing, Vol. 90, 106181, 2020, DOI: 10.1016/j.asoc.2020.106181.

 This work is licensed under a Creative Commons Attribution 4.0 International License